Design of Digital Systems Laboratory Exercise 4

Vending Machine

Océane Boulais
Performed: 10/9/2017
Submitted: 10/11/2017

Lab Section: 2
Instructor: Dr. Reza Azarderakhsh
TA: Amir Jalali

Lecture Section: Online
Professor: Dr. Reza Azarderakhsh

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: ______________________________

Table of Contents

I. Abstract

Although a vending machine is a very complex piece of machinery, the electronic controls are rather simple. FSM control of the vending machine is possible because the physical actions of the machine can be broken down to a small number of distinct states. These states are enumerated within the FSM and control is achieved by encoding the appropriate state transition Table 1 provided in the lab manual.

For this assignment, the model of the vending machine can be broken down into four modes: program (establish states), display (seven segment display), vend, and test (behavorial/ post-route simulations and hardware).

The action the vending machine will take in each mode is determined by a sequence of steps, each represented by a state that follows the logic provided in figure 1. There are a few other states which govern the flow of the table as well as some initial start-up states. It is a conglomeration of all these states that the FSM is built.

Code	Price
0000	$0.55
0001	$0.85
0010	$0.95
0011	$1.25
0100	$1.35
0101	$1.50
0110	$2.25
0111	$2.50
1000	$3.00
1001-1111	Reserved

***Table 1**: Control codes that correlate to soda prices*

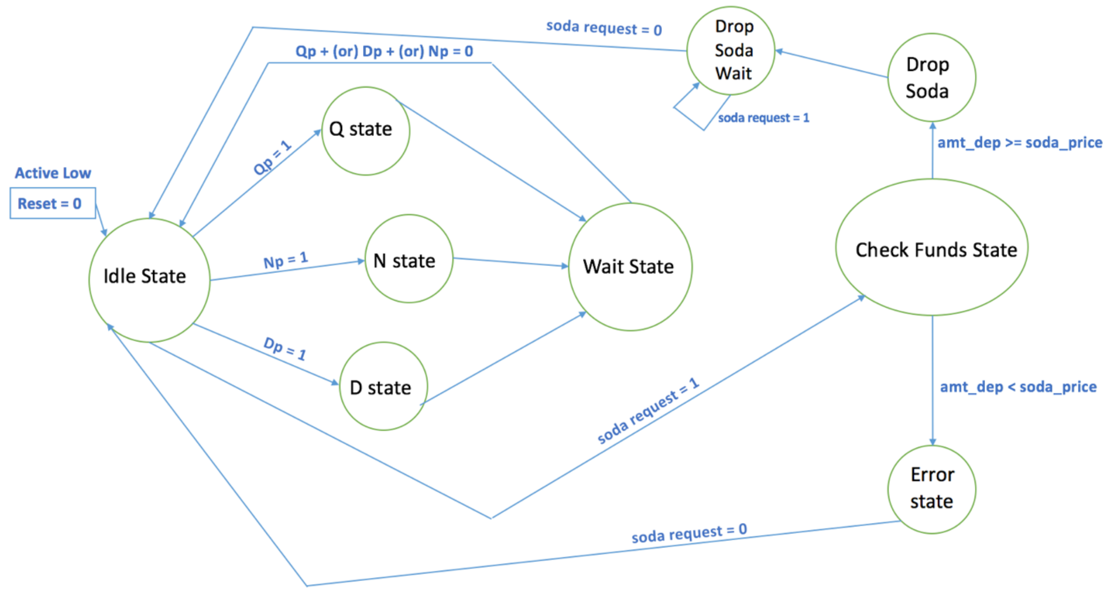

Figure 1*: FSM state logic (Prelab work)*

II. Design Methodology

For this assignment, the challenge was to build a coin controller that would handle the finite state machine and the incoming coins. Provided with the lab manual were the following VHDL modules: Usr interaction.vhd, bin bcd.vhd, coin rx.vhd, seven seg disp.vhd.
I was tasked with designing and implementing the coin controller, which would be the driver for state changes, the seven segment decoder to properly display amount deposited to the seven segment display. We shall now discuss these components and their design/implementation.

Vending machine controller

The vending machine controller encompasses the coin controller and seven segment decoder components. The vending machine is rising clock edge triggered and the reset is synchronous/active low.

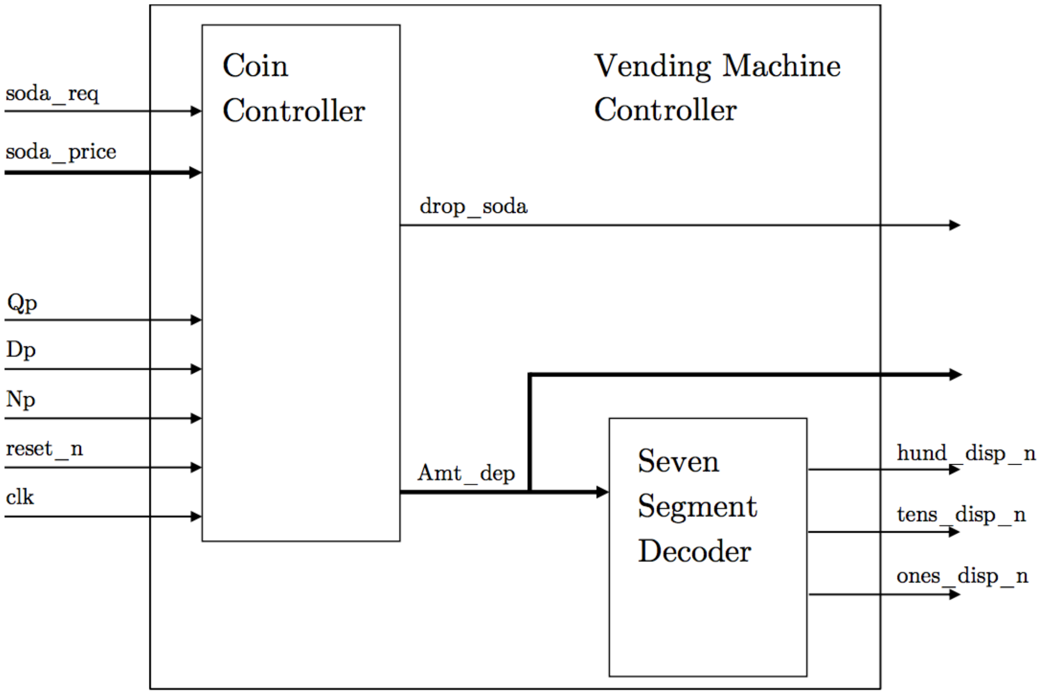

Figure 2: *Vending Machine High Level Block Diagram*

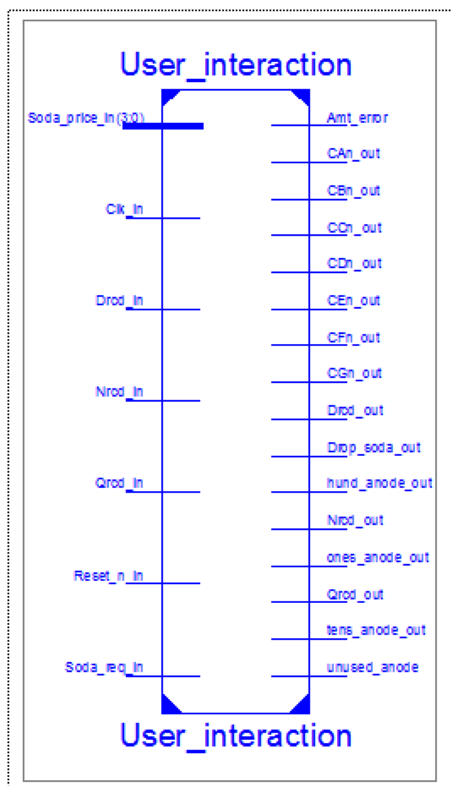

***Figure 3**: Vending machine RTL Schematic*

Coin Controller
The coin controller is where the finite state machine lives. By following Figure 1, one can see the abstract concept of changing and holding states. The system's state change is driven by a rising clock edge and active low reset. The user's input drives the internal state changes by the amount of coins put into the machine. The inputs were clock (rising edge sensitive), reset (synchronous, active low), Quarter, Nickel and Dime inputs, a soda request and a four-bit soda price (refer to Table 1).
The amount deposited does not clear any extra change after vending, instead, the change is kept and added to future purchases.

Seven Segment Decoder

The seven segment decoder was responsible for producing pulses that displayed the amount deposited on the seven segment display. The inputs are a 12-bit binary converted decimal, making a three-digit number capable of dynamic display on the seven segment display. The outputs of the decoder were the ones, tens and hundreds place digit.

User Interaction
The user interaction is the top-level of the vending machine. All created and given components are mapped and ported here. The following results shall clearly explain the current behavior of the user interaction.

III. Results

Vending Machine Controller
The first two sodas are requested here by their price code (0000) for simple demonstration of functionality. An amount error is triggered by a request for a soda with too little funds, and then the soda is vended once the funds have accumulated as show below in Figure 4. The amount deposit is given in binary representation for the segment display. At 235 ns, there is a $0.55 soda dropped. Binary to decimal conversions are provided for ease of analysis.

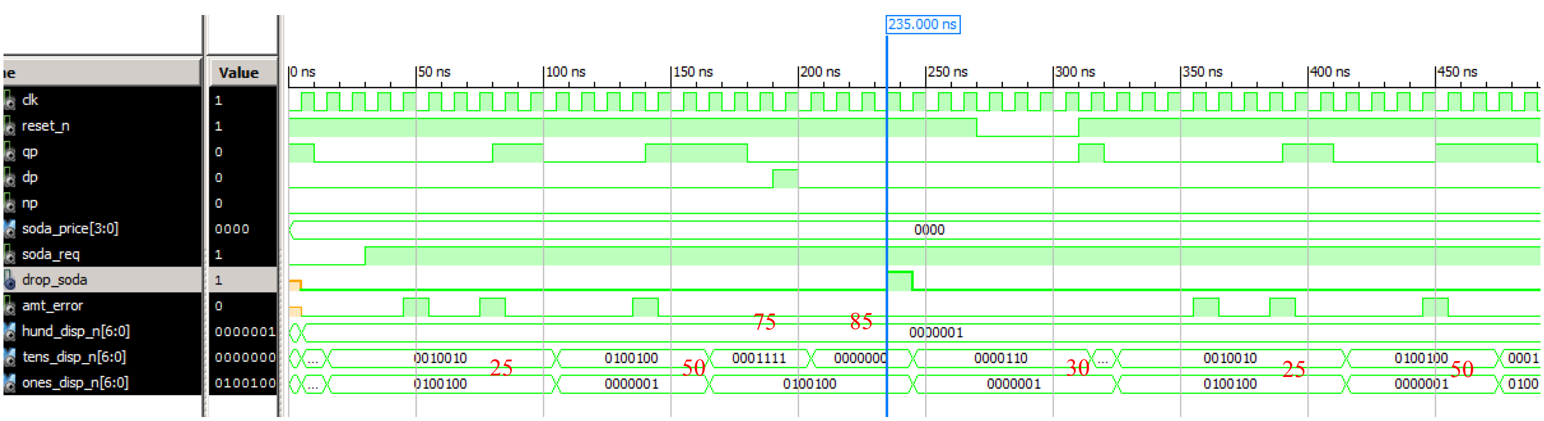

Figure 4a: *Vending machine behavioral simulation at drop soda*

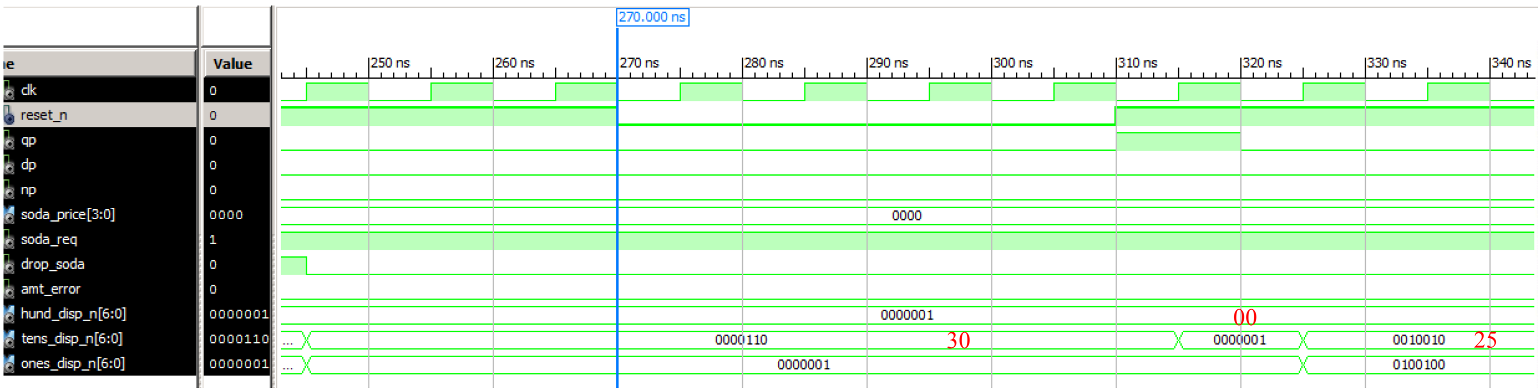

Figure 4b: *Vending machine behavioral simulation at reset*

Coin Controller

The heart of the vending machine controller, the coin controller shows the correlated amount deposited in decimal for easier viewing. One can also observe the "change" that is rolled over once a purchase is made. At 1.17 us, the $0.55 soda is dropped and the remaining change ($0.30) rolls over to next purchase. At approximately 1.35 us, reset is set to zero, therefore clearing whatever values were previously held in the amount deposited.

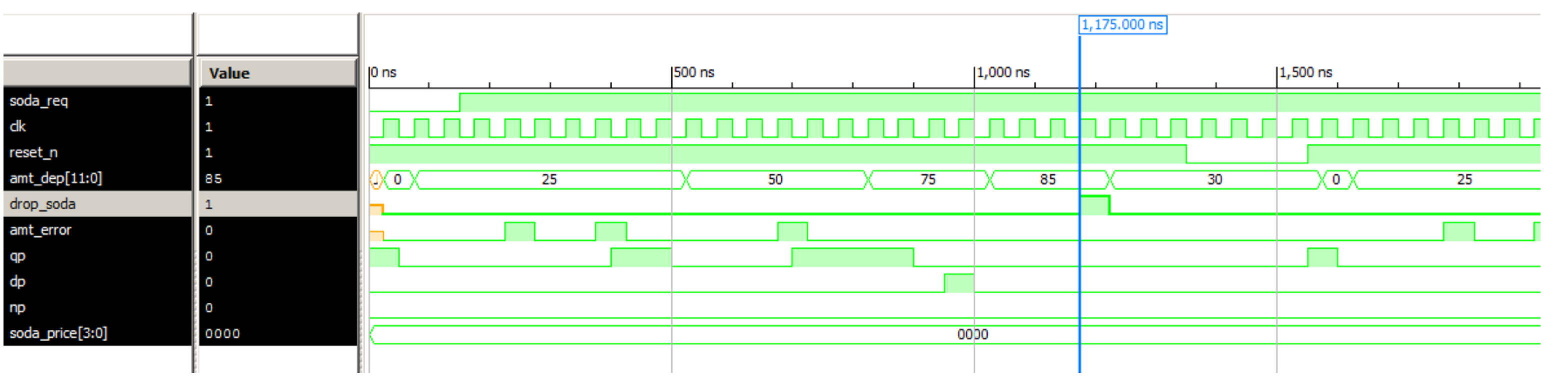

***Figure 5**: Coin controller behavioral simulation*

User Interaction

User interaction is provided as a top level of the vending machine. It connects the various components described above vis signal wires that later allow for behavioral simulation and post-route implementation. Table 2 gives an overview summary of the characteristics in the FSM.

States	7
Transitions	16
Inputs	4
Outputs	6
Clock	Clk (rising_edge)
Reset	Reset_n_INV_36_o (positive)
Reset type	synchronous
Reset State	idle
Power Up State	idle
Encoding	auto
Implementation	LUT

Table 2: Summary of FSM qualities

The user interaction was comprised of several components that were not built by the student. Although the user interaction should theoretically only be a giant port map, the components that were built and tested against respective test benches for this lab were unable to replicate the results from the vending machine and coin controller. Instead, the behavioral and post-route simulations match, and I implemented a successful run of the vending machine on the Artix 7. Potential solutions to continue debugging the error in the user interaction simulations include extensively following signal wire traces, re-wiring the input/outputs of the user interaction or simply build a new one from scratch.

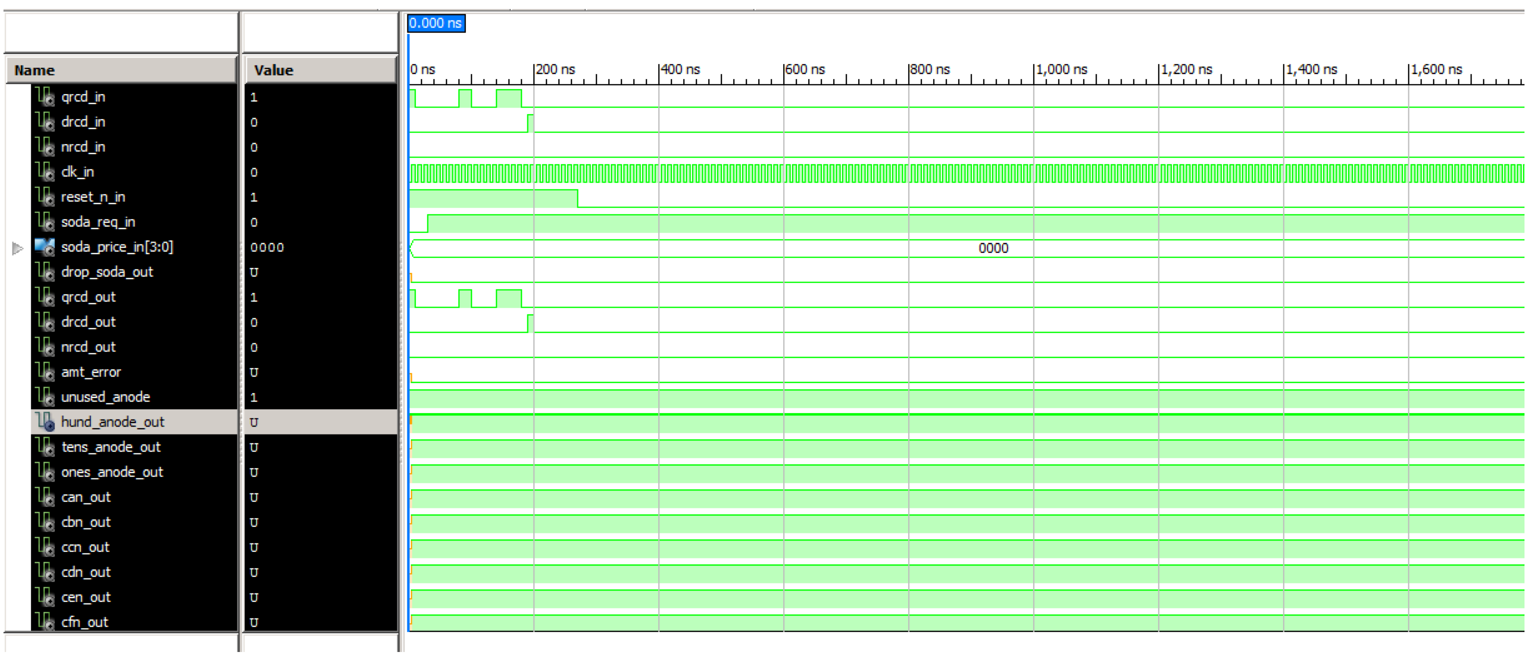

***Figure 6**: User interaction behavioral simulation*

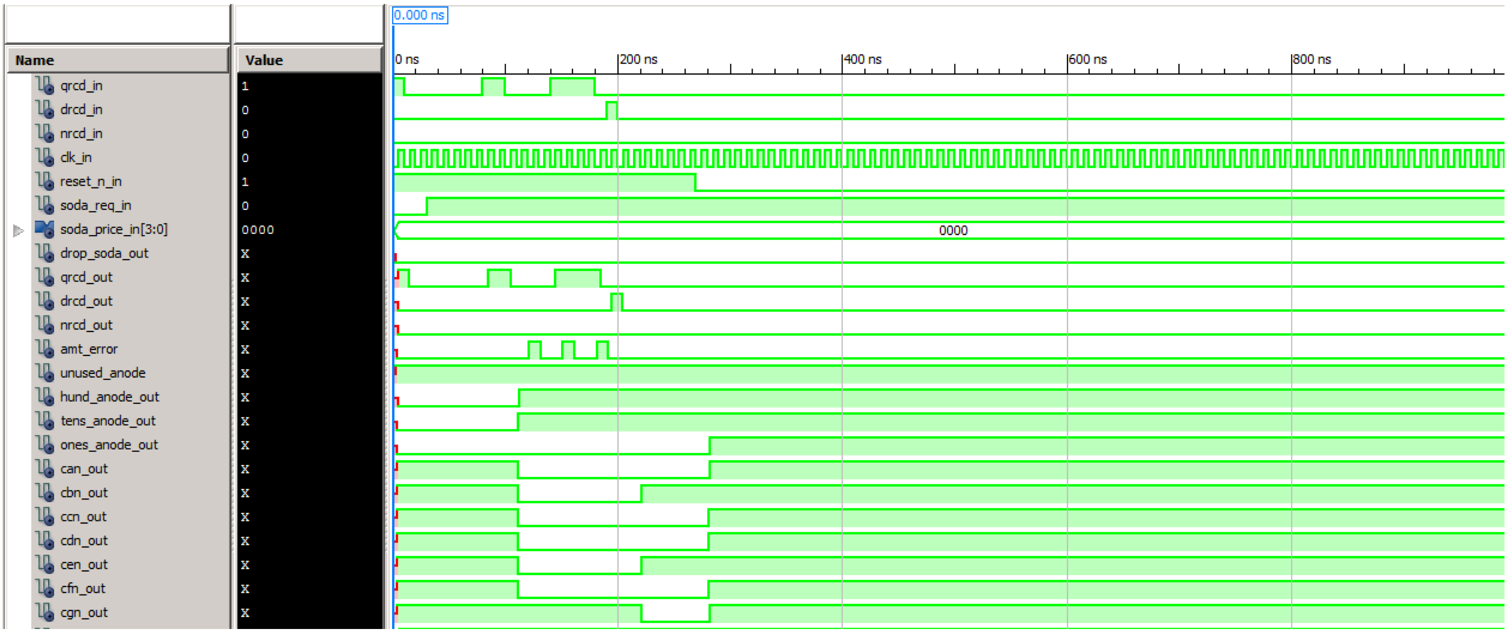

***Figure 7**: User interaction post-route simulation*

Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
Autotimespec constraint for clock net Clk_in_BUFGP	SETUP	N/A	6.105ns	N/A	0
	HOLD	0.030ns		0	0

Table 3: Best case achievable

Area Used

```
Device Utilization Summary:
Slice Logic Utilization:
  Number of Slice Registers:                   102 out of 126,800    1%
    Number used as Flip Flops:                  93
    Number used as Latches:                      9
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                        246 out of  63,400    1%
    Number used as logic:                      245 out of  63,400    1%
      Number using O6 output only:             193
      Number using O5 output only:              17
      Number using O5 and O6:                   35
      Number used as ROM:                    0
    Number used as Memory:                   0 out of  19,000    0%
    Number used exclusively as route-thrus:  1
      Number with same-slice register load:  0
      Number with same-slice carry load:     1
      Number with other load:                0
```

```
Slice Logic Distribution:
  Number of occupied Slices:                 92 out of  15,850    1%
  Number of LUT Flip Flop pairs used:       262
    Number with an unused Flip Flop:        164 out of     262   62%
    Number with an unused LUT:               16 out of     262    6%
    Number of fully used LUT-FF pairs:       82 out of     262   31%
    Number of slice register sites lost
      to control set restrictions:           0 out of 126,800    0%
```

IV. Appendix Page

Source Code Description

```
-----------------------------------------------------------------
-- File: usr_interaction.vhd
-- Entity:  User_interaction
-- Architecture: Behavioral
-- Author:  Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description: Top level wrapper of vending machine that provides port
mapping. Contains all following components.
----------------------------------------------------------------- -----
-- File: vending_machine_controller.vhd
-- Entity:  vending machine controller
-- Architecture: Structural
-- Author:  Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description: Contains coin controller and seven segment decoder.
Accepts quarters, nickels, dimes, soda requests and soda prices.
Outputs the seven segment display nodes that display amount deposit.
----------------------------------------------------------------- -----
-- File: coin_controller.vhd
-- Entity: coin controller
-- Architecture: Structural
-- Author: Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description: The coin controller is where a FSM resides along with a
series of prices that are used for soda selection.
-----------------------------------------------------------------
-- File: bin_bcd.vhd
-- Entity: binary vector
-- Architecture: Structural
-- Author: Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description: A function library that provides a function the will
```

```
convert a binary vector to a BCD vector
-----------------------------------------------------------------
-- File: coin_rx.vhd
-- Entity: coin rx
-- Architecture: Behavioral
-- Author: Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description: This component handles the debouncing of the buttons to
ensure that two coins are not counted when only one was inserted. As a
part of this, it outputs a signal for each received coin (via Qp, Dp,
and Np).
-----------------------------------------------------------------
-----------------------------------------------------------------
-- File: seven_seg_disp.vhd
-- Entity: Seven segment display
-- Architecture: Behavioral
-- Author: Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description: This module simplifies away those issues by taking in
the vectors from the seven segment decoder and outputting the necessary
signals for the display. Displays amount deposit.
-----------------------------------------------------------------
------------------------------------------------------------------
-- File: seven_segment_decoder.vhd
-- Entity: seven segment decoder
-- Architecture: Behavioral
-- Author: Océane Boulais
-- Create Date: 11/10/2017
-- Modified Date: 11/20/2017
-- Description:
The seven segment decoder is responsible for producing the pulses
necessary to display the amt dep on the seven segment display.
Inputs include the 12 bit BCD (three digit number in BCD)and the
Outputs:
– hund disp n (7 bits): Binary representing the segments that should be
on or off for the hundreds digit. Since the display has a common anode,
segments are active low.
– tens disp n (7 bits): Same as the hund disp n, but for the tens
place.
– ones disp n (7 bits): Same as the hund disp n, but for the ones
place.
-----------------------------------------------------------------
```

Waveforms

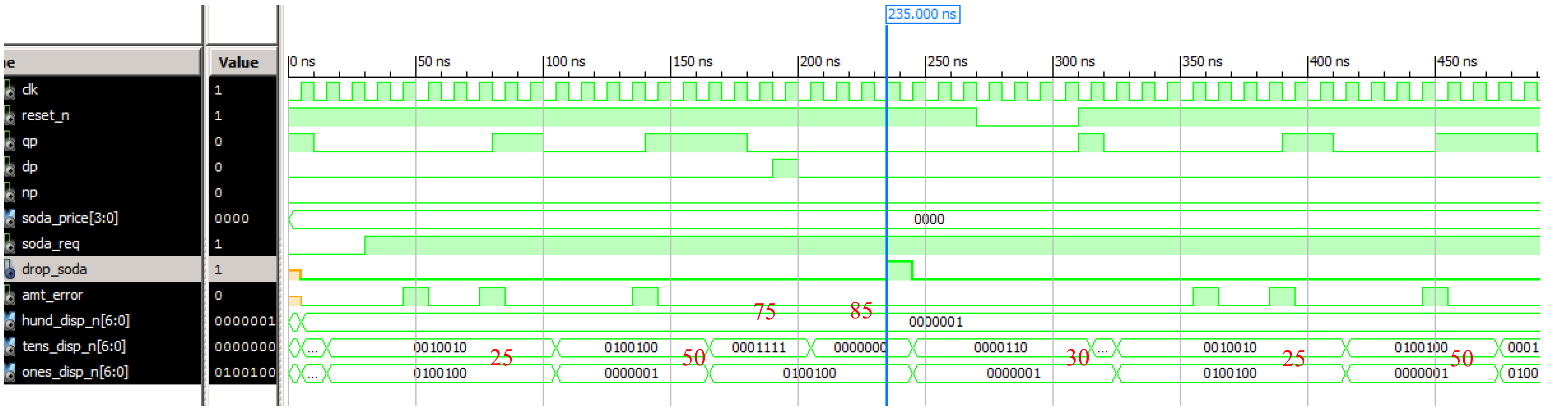

Figure 4a: *Vending machine behavioral simulation at drop soda*

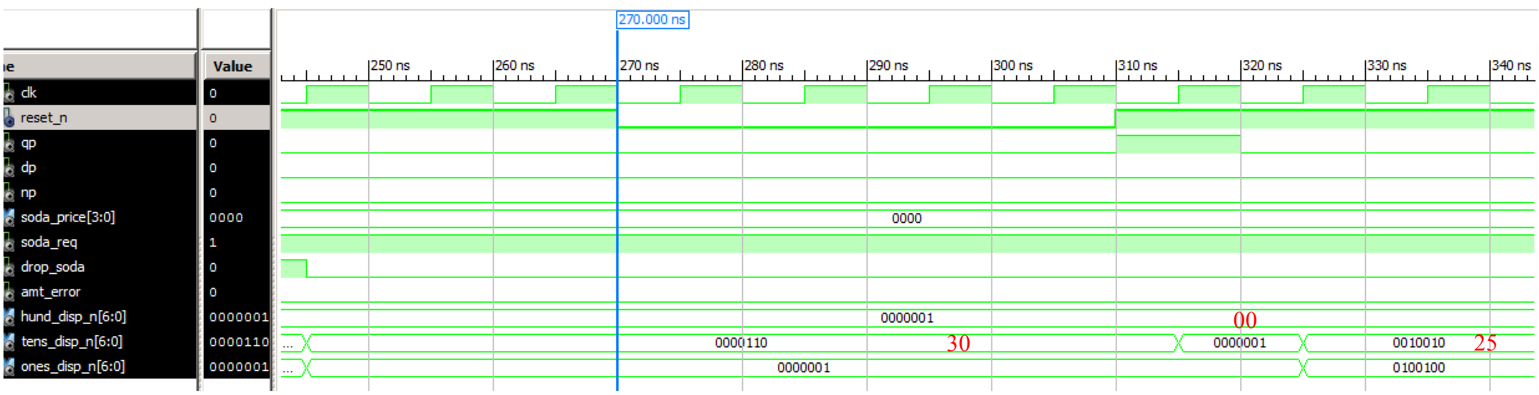

Figure 4b: *Vending machine behavioral simulation at reset*

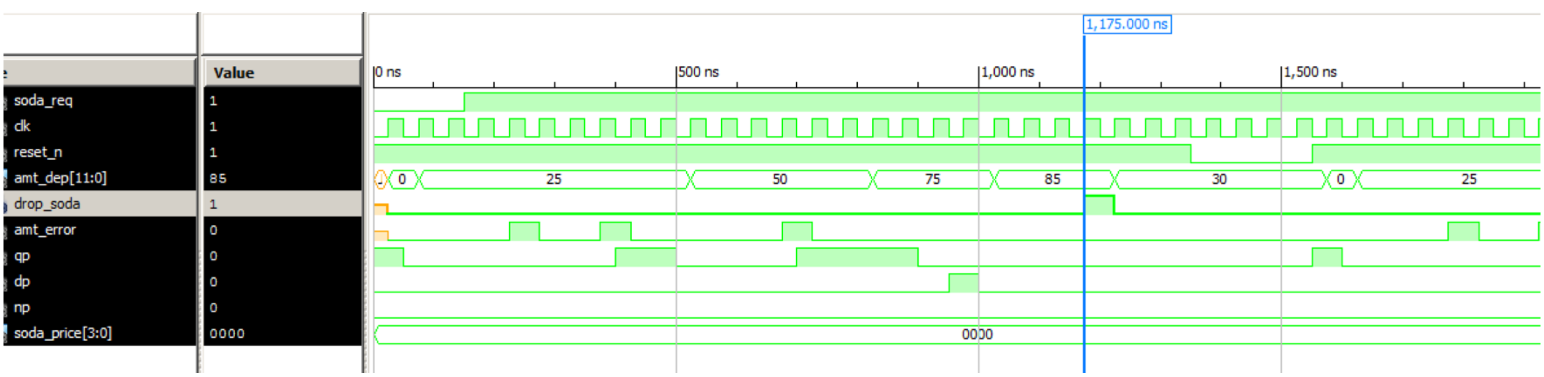

Figure 5*: Coin controller behavioral simulation*

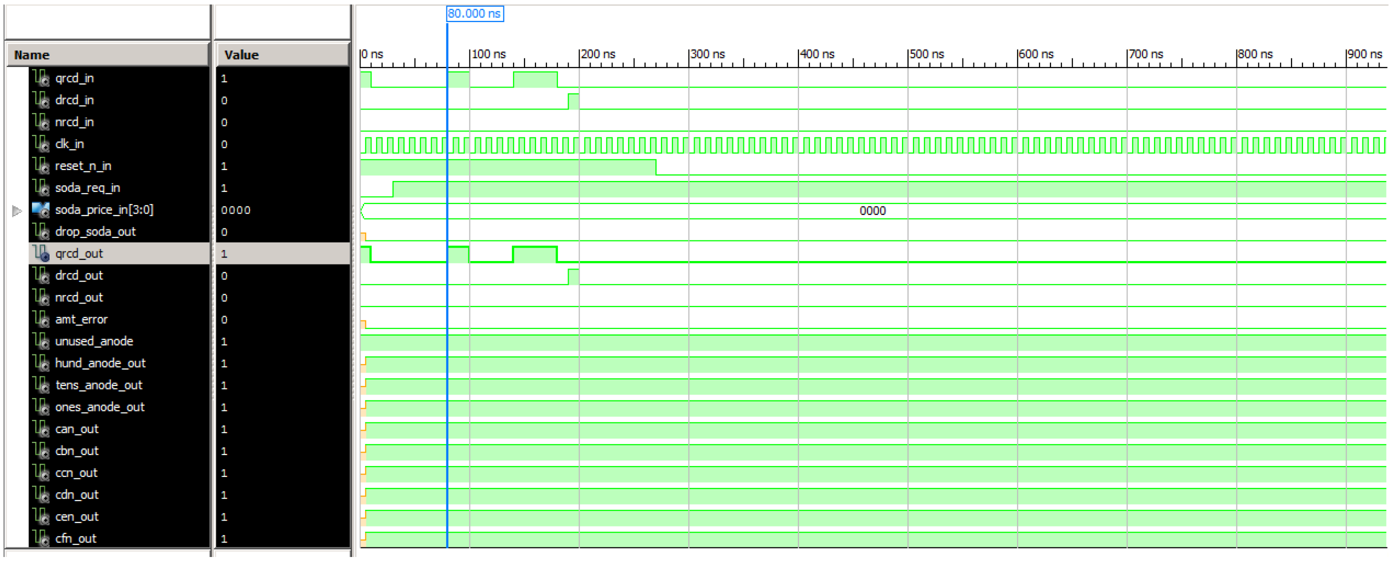

Figure 6*: User interaction behavioral simulation*

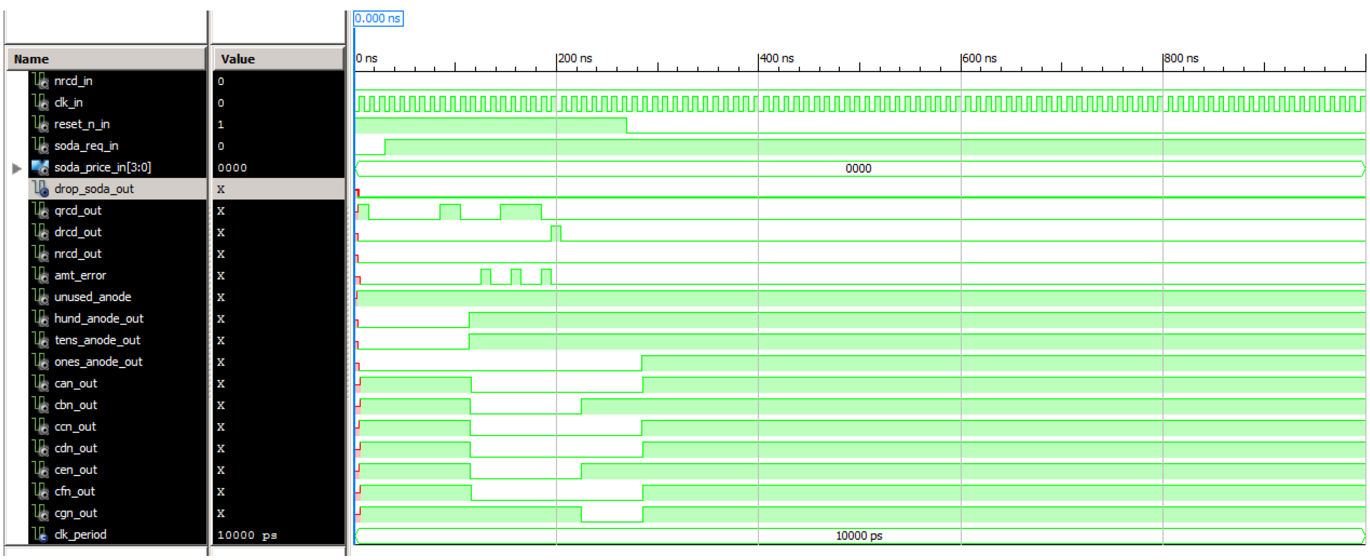

Figure 7*: User interaction post-route simulation*